Integrating IBM Maximo OMS and Canary Enterprise Solutions

Best-in-class enterprise historian solutions must deliver two key components, scalability and integration. Scalability however is about more than tag count; it also requires serving useful data to large numbers of clients. Likewise, integration is about more than connectors. Good database solutions must be able to contextualize data so that other systems can assimilate it properly.

FIELD NOTES: 2019 Midstream Oil and Gas Solution Recap

One of the largest midstream oil and gas companies in North America needed to better integrate their operation data with their IBM Maximo Order Management System (OMS). With more than 14,000 miles of pipeline to monitor and service, the undertaking felt massive. A successful solution would require high scalability and an extremely flexible architecture for third-party software integration.

The Canary System was chosen after extensive research and a thorough RFP process. After deployment and data logging was completed, more than 560,000 tags were archived in the enterprise level Canary Historian. With scalability almost immediately demonstrated, providing integration with the Maximo OMS was the next challenge to complete. However, raw tag data would not be sufficient for the needs of the organization. To provide actionable information, the Canary data had to be contextualized.

To achieve this, a master tag mapping table was constructed in MSSQL and contained known SCADA/DCS tags, their matching Canary tag counterpart, and any data within Maximo that would be useful to associate as metadata properties to the Canary tag within the Canary System. This metadata included the known Maximo Asset ID Number (ASSETID) and would include data regarding service history, manufacturer information, and other key properties. The newly defined master tag table would remain dynamic, allowing for updates when new tags or new asset data from Maximo were introduced.

Leveraging Canary’s ability to monitor external databases for metadata properties, the metadata within the master table can then be imported into the Canary Views service. Since Views acts as a single endpoint for all client queries, all Canary Historian tags can have these added metadata properties exposed for additional contextualization by both Canary and third-party applications.

Already seven Canary Virtual Views had been built to power other reporting operations and data integration processes. For Maximo integration, it was decided to build an additional Virtual View, specific to the unique needs of the Maximo integration.

The dedicated Maximo view would focus on representing the pipeline's critical asset infrastructure. To complete the task, more than 80 Canary DataSets were joined, totaling more than 135,000 tags. Assets were created with multiple levels of parent-child hierarchy, identifying more than 30 different asset types. Compressors proved to be the largest total category, featuring more than 330 unique instances distributed across the 14,000 miles of infrastructure.

Once the Virtual View and asset model were completed, the Canary Calculated Server was leveraged to provide key calculated data points for each asset. After a calculation was defined, it was deployed across all known instances of the linked asset type. The Maximo Virtual View automatically discovered the new calculated tags and made them available for client queries.

These calculated data points included various runtime and downtime metrics which would be used within Maximo. In addition to runtime, the calculation server could also be used to build complex logic expressions for condition-based asset monitoring triggers. These calculated Booleans would become indicators of whether an asset is, or is not, in an event state based on the defined expression.

To incorporate this data into Maximo, the Canary ODBC Data Connector was used, allowing for SQL-like queries to be made against the Canary Views. This was extremely useful for pulling asset tag values out of Canary and inserting them into existing Maximo MSSQL tables. Using the master tag lookup table and linking Maximo ASSETID to the Canary asset instances ensured all runtime and downtime metrics were properly linked to their Maximo asset counterparts.

In addition to serving Maximo data from the Canary System, further integration was still possible. Most desired by the pipeline were for work orders to be automatically generated based on the calculated Boolean tags monitoring assets for certain conditions. To achieve this outcome, the Canary Event Service would trigger a new event when a calculated Boolean changed to nonzero. When this occurred, user defined analytic data would be captured based on the duration of the event and all collected Canary Event data could then be written to a Maximo MSSQL instance.

Additional reporting integration between the Canary and Maximo system was achieved by calling the Maximo ASSETID metadata property linked to each Canary asset instance. Within the Canary dashboarding and visualization tool Axiom, all known asset instances of the 'Air Compressor' asset type were visualized and filtered/sorted based on tag values. In addition, the ASSETID was exposed. Each air compressor instance within the Axiom HTML dashboard could then include an external hyperlink to Maximo which passed the ASSETID into the URL, providing 'one-click' access to the exact asset instance within the Maximo application.

Canary implementation for a similar project of this scope can be accomplished very rapidly, and scale to any size enterprise, both in tag count and in integration needs; especially when compared to other historical database solutions. The bulk of implementation work for the project revolved around data logging, Virtual View build out, calculation creation, and Axiom screen development.

In particular, the Maximo integration could be replicated in roughly eighty manhours, assuming the master tag table had already been defined. The project could require significantly more energy depending on the complexity of the work order generation requirements.